CHECKERS in the last issue had one typo; in line 1220 where part of the line read:
            @((S+R)+2)=3;   and it should have read: @((S+R)÷2)=3;
An error in line 8 had too many zeros at 30000.
I've had many comments on this program, all pleased with the effect and operation
(once the glitch was cleared up).

PROOFREADING of listings is getting better, and hopefully I'm not going to have
any more problems as I am now sending a copy of the ready-to-print listing to the
author for his last minute check.

GIANT LETTERS program on p.45 can be 'clarified' by replacing line 105 with the
statement X=0; GOSUB C . The zero will stop the machine's printing after it
finishes the AN of ARCADIAN. What is happening is - the machine has been set into
a printing mode and it keeps on going until it hits an internal halt. The X=0 sets
such a halt thru the POKE function, where you want it.

CONNECTOR for the 50-pin expansion port is a standard printed circuit device, with
25 double-sided pins on 0.1 inch centers. I understand that the APPLE II uses this,
as well as the SHUGART disc drive. There are some of these on the surplus market
here, 3M No. 3415, and I can supply these at $2.50 ppd, including a 3" bit of flat
cable that you could solder to.

KEYBOARD project in work utilizes a CHERRY brand (?) keyboard wired in parallel
with the keypad with some buffers so that either can be used. All letters and
characters are where they belong, while the shift key allows the generation of the
various WORDS. The Keyboard is the one advertised by JAMES ELECTRONICS, 1021
Howard Ave., San Carlos CA 94070, at $29.95. "63 Key Uncoded Keyborad".

PRINTER project has been made to work ,using a Type 43 Teletype machine and a kit
for an interface available from ELECTRONIC SYSTEMS, P.O. Box 21638, San Jose, CA
95151, their part #232A at $7. plus postage. I saw an ad of theirs in KILOBAUD,
April, p.172. The schematic of this kit is included for you scratch-builders:
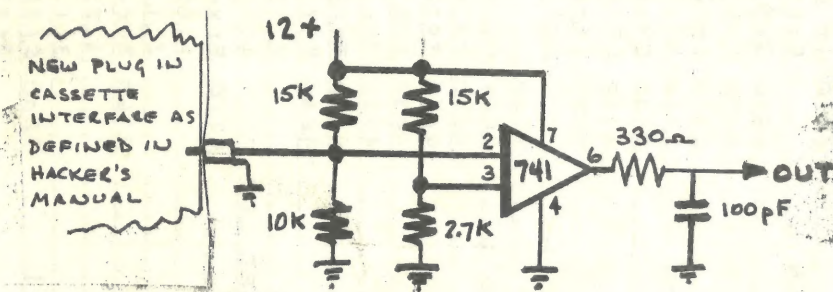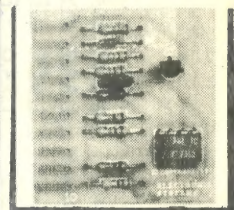
COLOR STANDARDS question came up, and I find that Bally would make an excellent
color generator, as apparently their color output is very well controlled. I
wonder if one of the TV-technical types can compare colors with a standard color
generator output and tell us which color numbers give you the standard shades
necessary for color TV work.

BANGMAN game program included this month is by Ernie Sams,248 S. Forest Street,
Bellingham WA 98225. It has a novel twist over the old hangman game and is quite
clever. It has a good scheme for entering characters without their appearing on
the screen, and a search routine that can locate and account for multi-usage of
a letter. I am also including Ernie's sheet of documentation that will be of help
to a lot of us.

Line #	Statement(s)
3	.HANGMAN.
5	.BY E.SAMS (C).3-3-79.
7	
9	NT=2
10	CLEAR;PRINT"ONE PERSON.
	KEYS IN A WORD TO 10 LET-
	TERS;ANOTHER TRIES TO
	GUESS IT WITH NO.
20	PRINT"MORE THAN 9 WRONG
	GUESSES USING KN & TR (1)
30	PRINT;PRINT"PRESS GO.
50	I=KP;IF I=13 CLEAR
100	E=0;Q=0
200	PRINT"ENTER WORD, THEN
	PRESS GO.
300	FOR A=0 TO 9
310	CY=31
320	PRINT"LETTER #",#3,A+1
330	BOX-55,23,50,8,2
400	K=KP
410	IF K=13 GOTO 790
420	IF K>90 PRINT"INVALID";
	GOTO 310
430	IF K<65 PRINT"INVALID";
	GOTO 310
600	@(A)=K;@(A+10)=K
700	NEXT A
710	CLEAR
800	FOR B=0 TO A-1
900	Y=-32;X=-80+(B*8)+3
1010	BOX X,Y,7,3,1;NEXT B
1100	CY=-24;CX=6;PRINT"USED LET
	TERS";CY=-32;CX=42;
	PRINT"BELOW:
1200	FOR M=0 TO 25;@(M+20)=
	M+65;NEXT M
1300	FOR D=0 TO 25
1400	CY=3.1;CX=-80
1410	PRINT"GUESS #",#3,D+1
1420	BOX-33,19,94,16,2

Line #	Statement(s)
1430	G=(KN(1)+132).÷10+65
1440	CY=23;CX=-74;TV=G
1450	IF TR(1)=0 GOTO 1430
1540	IF@(G-45)=-1 PRINT"HAS
	BEEN USED",;GOTO 1400
1550	@(G-45)=-1;CY=-40;CX=
	-78+((G-65)*6);TV=G
1600	FOR C=0 TO A-1
1700	CY=-25
1800	IF G=@(C) CX=-79+(C*8)+3;
	TV=G;Q=1;@(C)=-1
1900	NEXT C
2000	IF Q#1 GOSUB 9600+(E*10);
	E=E+1;IF E=9 GOTO 9000
2050	Q=0
2100	FOR F=0 TO A-1
2110	IF @(F)#-1 GOTO 2200
2120	NEXT F
2130	CY=17;CX=-50
2140	PRINT"CONGRATULATIONS
2150	CY=0;CX=-80;PRINT"PRESS
	GO";GOTO 50
2200	NEXT D
9000	CX=-80;CY=-9
9010	PRINT"SORRY ABOUT THIS.
9020	PRINT"THE WORD WAS
9030	FOR H=0 TO A-1
9035	CX=-79+((H*8)+3)
9040	TV=@(H+10);NEXT H
9060	BOX-16,0,10,10,1;BOX-8,7,
	27,5,1;BOX-19,1,1,2,1;BOX
	4,1,1,2,1;BOX-9,4,4,1,4,1
9110	BOX-6,3,1,5,1;BOX-8,9,6,1,
	1;BOX-18,7,3,3,2;BOX-17,7,
	2,1,1;BOX-14,7,3,3,2
3160	BOX-14,7,1,1,1;BOX-11,7,1,
	3,2;BOX-9,6,2,1,2;BOX-7,8,
	3,1,2;BOX-7,7,1,1,2,2
9200	&(9)=222;&(0)=87;&(1)=87;
	&(2)=80;&(3)=80

Line #	Statement(s)
9225	FOR U=0 TO 5
9230	&(23)=255; &(21)=255
9240	&(23)=0; &(21)=31
9260	FOR Y=5 TO 20;&(19)=Y;NEXT Y
9270	&(21)=0; &(19)=0
9300	X=RND(10)+55
9310	Y=RND(12)
9320	BOX X,Y,1,1,2
9340	NEXT U
9350	FC=80; &(10)=0
9360	BOX 60,10,38,60,2
9370	BOX-8,4,30,18,2
9400	FOR Y=0 TO 176;&(10)=Y;NEXT Y
9410	&(9)=50; FC=0
9500	CX=-50;CY=16;PRINT"PRESS.
	GO"; GOTO 50
9600	BOX 60,27,10,14,1;BOX 58,
	28,2,1,2;BOX 62,28,2,1,2;
9610	BOX 60,24,6,1,2;RETURN
9620	BOX 60,18,4,5,1;RETURN
9630	BOX 60,6,16,20,1;RETURN
9630	BOX 50,14,4,3,1;BOX 49,6,
	3,3,20,1;RETURN
9640	BOX 70,14,4,3,1;BOX 70,6,
	3,20,1;RETURN
9650	BOX 57,-12,3,15,1;RETURN
9660	BOX 64,-12,3,15,1;RETURN
9670	BOX 55,-20,8,2,1;RETURN
9680	BOX 67,-20,8,2,1;RETURN

NOTE: ELIMINATE AS MANY SPACES
AS POSSIBLE(EXCEPT THOSE
INSIDE QUOTES).IF YOU RUN
SHORT,CUT LINES 3-30

BANGMAN DOCUMENTATION:

10 - 200	Initialize and instructions.
300 - 700	Ask for and accept up to ten valid letters.
600	Stores the word in two locations:
	1. To keep track of the letters correctly guessed.
	2. To print the word if not guessed in nine tries.
800 - 1010	Set up blanks for the word.
1200	Store each letter of the alphabet for future use so the same letter is not used twice.
1300	Initiate guessing loop. Allows 26 guesses.
1430 - 1450	Allows one to guess a letter by turning knob #1.
1540	If storage position is set to -1 the letter has been used.
1550	Sets value of storage position to -1 and prints letter at the bottom of the screen.
1600 - 1900	Loop through the storage positions in 600 (1). If a match
1800	is found print the letter in the appropriate location(s) on the blanks established in 800-1010 and change the storage position value to -1. Set flag "Q" to 1.
2000	If the flag "Q" is not equal to 1 then the chosen letter did not match a letter in the word so go to subroutine 9600 plus counter E times 10 and print that portion of the man. Increment the E counter. If there have been nine wrong guesses default to 9000 to "bang" part of bangman.
2050	Otherwise flag "Q" equals 0.
2100 - 2200	Loop through storage positions in 600 (1). If all positions are -1 then the word has been guessed. Print "congratula-tions" and press go to start a new game.
9000 - 9020	The word was not guessed within the nine wrong guesses allowed. The man was completely built, so - -
9030 - 9040	Print out the word (from 600 (2)) on the blanks.
9060 - 9160	Draw a gun with the word COLT on it.
9200	Change the screen to a border format.
9225 - 9340	Put six random shots in the body of the man. Use sound effects. 9230-9240 is the shot. 9260 is the ricochet.
9350	Blank out screen
9360 - 9370	Blank out man
9400	Uncover screen from top to bottom with man gone.
9410	Restore screen to full screen format.
9500	Press go to start a new game.
SUBROUTINES	
9600	Prints head, eyes, mouth.
9610	Prints neck.
9620	Prints body.
9630	Prints right arm.
9640	Prints left arm.
9650	Prints right leg.
9660	Prints left leg.
9670	Prints right foot.
9680	Prints left foot.

RANDOM ART has been expanded with some added sound and shape enhancements by Dave Stocker. Add the following to last month's program:

```
 15 &(Ø) =7; &(1)=7; &(9)=84; NT=Ø,&(21)=14; &(22)=255
 85 B=FC+RND(31)x8+4; &(2)=B; &(3)=B
125 &(19)=X; &(18)=Y
135 IF &(23)=1 RUN
```

Press GO to set new parameters.   (try 5,13)

TUTORIALS by Steve Walters (556 Langfield, Northville, ME 48167) and Dave Iback (19553 Dartmouth) follow. The first provides some very interesting basic data on character size and how to make them appear exactly where you want them. The second provides additional comments on the IF-type statements discussed on p. 41.

CHARACTER SIZE AND PRINT LOCATION

Every character is 5 pixels wide, with a one-pixel space added to the right of the character to provide a one-pixel separation between characters. Thus, the effective width of a character is 6 pixels.

Every character is 7 pixels high, with a one-pixel space added beneath it to provide a one-pixel separation between lines of print. Thus, the effective height of a character is 8 pixels.

The cursor is displayed as a 6-pixel wide by 8-pixel high box, the effective size of a character. When a program is running, the cursor is not visible. When a program ends, the cursor is displayed wherever it happens to be, preceeded by the line entry indicator ⟩ .

If a PRINT command is not ended with a comma, the computer will leave a full space (6 pixels by 8 pixels) following the last character in the PRINT statement, and shift to the next print line. This end-of-statement space will appear as a white box against a black background, and will blank out anything located beneath it.

If a PRINT command is ended with a comma, no space is added beyond the one-pixel space to the right of the last character, and the cursor remains at that location until another PRINT command is given.

A character prints centered on its CY location, but not centered on its CX location. Given CX and CY as the print location of a character, the horizontal center of the character is CX-1 and the vertical center of the character is CY. The left edge of the character is located on CX-3 and the right edge of the character is located on CX+1. The top and bottom of the character are located on CY+3 and CY-3, respectively.

Since the left edge of the screen display area is on CX=-80, and the left edge of a character is on CX-3, a character will print at CX=-77 even if the program specifies CX=-78, -79 or -80 prior to the PRINT command. However, this behavior is not duplicated at the right side of the screen. If CX is specified at +78, the right edge of the character will be on CX=+79 (CX+1, and the right limit of the screen display area); but if CX is specified at +79, the character will print beyond the CX=+79 limit. The cursor will shift lines in the process, and if a comma follows the PRINT command, the cursor will shift to CX=-77 on the same CY line.

SCREEN DISPLAY AREA RELATED TO CHARACTER SIZE

The edge limits of the screen display area (CX=-80 to +79; CY=43 to -44) are functionally related to the character print size. The normal top line of print (without a CY value being specified) is CY=40, and thus the top of the characters on that line are at CY=43. There are 11 normal print lines, located at CY=40, 32, 24, 16, 8, 0, -8, -16, -24, -32 and -40. The bottom line (CY=-40) results in the bottom of the characters on that line being at CY=-43 and the one-pixel space below them being at CY=-44.

Similarly, the screen display area width is 160 pixels. This would allow 26 characters (26x6=156 pixels) with 4 pixels to spare. However, the first line of an entry starts with the line entry indicator) as the first character, and so only 25 characters and spaces can be entered. When 25 additional characters have been entered, the one-pixel space to the right of the last character is at CX=75: an additional character cannot fit in the remaining 4 pixels so the cursor shifts to the next line. On second and subsequent lines of an entry, 26 characters can be entered per line.

When a program is being LISTed, the address of each line entry starts 2 characters to the right of CX=80, at CX=62. Thus, the first line including address will contain 24 characters and spaces; and the second and subsequent lines will contain 26 characters and spaces.

BOX COMMANDS RELATED TO PRINT LOCATION

The dimensions of a box which will outline a printed statement, or blank it out, can be readily determined from the CX and CY values of the PRINT command.

Framing a printed statement or blanking it out with a reverse box. The smallest box which can frame a character is a reverse box:

    BOX CX-1, CY, 7,9,3

The above box is a rectangle which is 7 pixels wide and 9 pixels high superimposed over the 5 pixel by 7 pixel character.

A more esthetically pleasing reverse box frame is obtained from a square box:

    BOX CX-1,CY,9,9,3

A similar formula can be used to blank out a character:

    BOX CX-1,CY5,7,2

Note that the reverse box must be produced after the PRINT command for the statement being framed or blanked out has been executed.

Framing a printed statement with an outline box. The outline box X is more useful than the reverse frame box because the outline box can be produced before the PRINT command is executed, and thus the character can be printed and reprinted inside the outline box as desired during a program.

The smallest outline box which can frame a character is:

    BOX CX-1,CY,9,11,1
    BOX CX-1,CY,7, 9,2

Again, a square outline box is more esthetically pleasing:

    BOX CX-1,CY,11,11,1
    BOX CX-1,CY, 9, 9,2

Outline box for a statement of more than one character. Given CX and CY as the PRINT start location for a row of "n" characters (n=1 or more), with a comma the outline box which will frame the row of "n"

characters can be determined from the following:

    BOX CX-1+3(n-1),CY,11+6(n-1),11,1
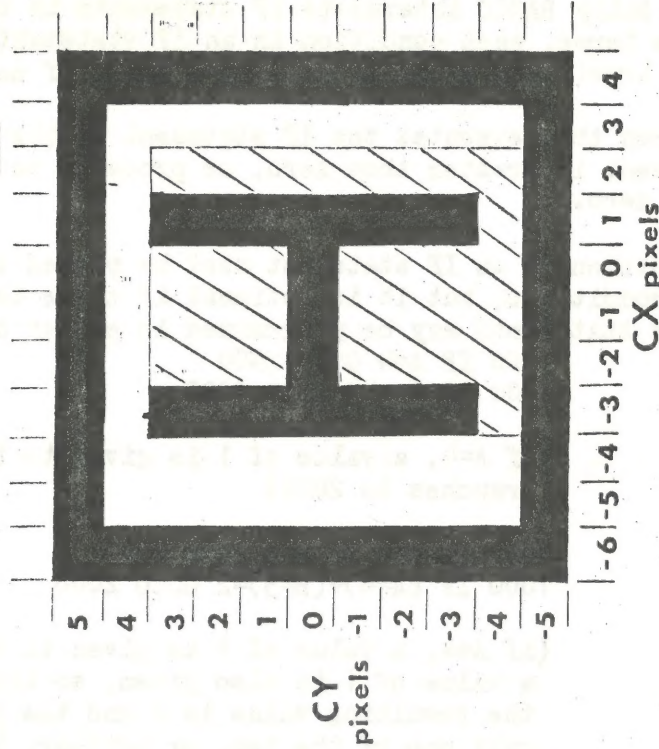   BOX CX-1+3(n-1),CY, 9+6(n-1), 9,2

Note: a reverse box could also frame the row of characters using either of the above BOX commands with a 3 as the ending code.

If a comma is not desired to follow the PRINT instruction, then a larger outline box is required because of the 6-pixel space added after the last character:

    BOX CX-1+3(n-1),CY,21+6(n-1),11,1
    BOX CX-1+3(n-1),CY,19+6(n-1), 9,2

Illustration of an outline box and a printed character inside it:

    BOX CX-1,CY,11,11,1
    BOX CX-1,CY, 9, 9,2
    CX=0; CY=0; PRINT "H",

Area cleared when "H" is printed

TUTORIAL 2 is a follow-up on last issue's comments on IF statements, again by Dave and Steve.

The Bally BASIC interprets IF statements in terms of Boolean algebra concepts. In simple terms, each condition in an IF statement is assigned a value of one (+1) if it is true(i.e., if it is met) or a value of zero ($\emptyset$) if it is false (not met).

The program then executes the IF statement if the resulting Boolean value of the IF statement is greater than zero, or proceeds to the next program line if the value is zero.

Each condition in an IF statement must be placed in parentheses if there are two or more conditions, but it is optional if there is only one condition. Thus, simple IF statements may be programmed in either of two ways:

```
1000 IF A=4 GOTO 2000                    (uses 12 bytes)
1000 IF (A=4) GOTO 2000                  (uses 14 bytes)
```

(if A=4, a value of 1 is given to the IF statement and the program
  branches to 2000)

```
1000 IF A=4 IF B=5 GOTO 2000             (uses 16 bytes)
1000 IF (A=4)+(B=5)=2 GOTO 2000          (uses 22 bytes)
```

(if A=4, a value of 1 is given to the IF statement, and if B=5,
  a value of 1 is also given, so that if both conditions are true
  the resulting value is 2 and the program branches to 2000; if
  only one of the two, or neither, is true, the program does not
  branch)

Obviously, in both of the above cases, using the parentheses costs more in memory bytes and is thus not efficient. However, in more complicated IF statement conditions, parentheses result in less memory bytes being used; and in some cases the saving is substantial.

For example, consider a condition where A=4 or B=5 or both (i.e.,A=4 and/or B=5);

```
1000 IF A=4 GOTO 2000
1010 IF B=5 GOTO 2000                    (uses 24 bytes)

1000 IF (A=4)+(B=5) GOTO 2000            (uses 20 bytes)
```

Even more memory savings would be involved if the condition was any one or more of three variable values.

The following list illustrates other applications of special IF statements in the most byte-efficient form, based on the above principals:

Special IF statements:
```
If A=1 and/or B=2          1000 IF (A=1) + (B=2) GOTO 2000
If A=1 and B=2             1000 IF A=1 IF B=2 GOTO 2000
If A=1 or B=2 but not both 1000 IF (A=1) - (B=2) GOTO 2000
If A ≠ ∅ (i.e., if ABS (A) ∅) 1000 IF A GOTO 2000
```
         This is also useful for hand control responses:
```
1000 IF TR(1) GOTO 2000        will branch if trigger is pulled
1000 IF JX(1) GOTO 2000        will branch if joystick moved sideways
1000 IF JY(1) GOTO 2000        will branch if joystick moved back & forth
```
```
If neither A nor B equal zero  1000 IF A IF B GOTO 2000
If A=1 and B=2 and C=3         1000 IF A=1 IF B=2 IF C=3 GOTO 2000
If A=1 and/or B=2 and/or C=3   1000 IF (A=1) + (B=2) + (C=3) GOTO 2000
If A=1 or B=2 or C=3 but not more
   than one of these          1000 IF(A=1) + (B=2) + (C=3) =1 GOTO 2000
If any two of the above three
   conditions are to be met    1000 IF(A=1) + (B=2) + (C=3) =2 GOTO 2000
```

TUTORIAL 3 is another follow-up, this time by Jean Taillefer.
     The $>=$ and $=<$ symbols will work in an IF statement. They stand for
                    "equal to or greater than" and "less than or equal to",
respectively. The symbols are sometimes shown as $\geq$ or $\leq$.
     The statement IF A$>$=B GOTO 100 is equivalent to
          IF A$>$B GOTO 100;IF A=B GOTO 100   or
          IF(A$>$B) + (A=B) GOTO 100

     A statement that is equivalent to the 'reverse sign' function on a calcu-
lator can be accomplished with the following: -(A)
     If A is positive, the statement is evaluated as -(+A),for a negative number.
     If A is negative, the statement is evaluated as -(-A),for a positive number.
     This statement can be used with a variety of other statements, such as:
               PRINT -(A)
               B=-(A)
               IF -(A) = 1 GOTO 100
     This is different from the ABS command which always gives a positive number.

DEALER in the St. Louis area is RTS Electronics/Gametronics, 356 Brookes Dr.
731-2309 who report that they have all cartridges in stock, do mail order, and
hold Football tournaments for players on Tuesdays,6-9pm and Saturdays, noon to 4.
They mention that players are bringing their own hand controllers, which makes me
think of the slot car operators...

DEALER in Indiana is ABC Hobbycraft 2155 E.Morgan Ave,Evansville 47711 (477-9661)
who are open Sundays, and who act as the local hangout for Bally enthusiasts.

GAMES 2007 PINBALL and 2006 SPACE INVADERS are scheduled for July release. There
were a few words on these on p. 23.

STATEMENT When I started this paper late last year, I indicated in my subscrip-
tion form that I would print material as it became available, but at least bi-
monthly in this fiscal year. Well, I have been doing better than bi-monthly because
material has been coming in at a much higher rate than I expected, so I have been
publishing more or less monthly. This is a hobby output so publication is bound to
be somewhat haphazard. Post-printing operations are starting to get unwieldy and I
may have to get professional help in the folding/stapling area. Right now it looks
like there will be at least 2 more issues to November, so you'll be getting at
least a 50% 'profit'.

<u>ADS</u>   Sell: Bally ARCADE with 4 hand controls,280ZZZAP,Baseball,Tiny BASIC. First
certified check for $260. R.Dermody,8431 Timber Glen,San Antonio,TX 78250

A fantasy game package is available for those who enjoy the Dragon/Dungeon
type of operation. G.McLimore,1210 E.Virginia St. Evansville IN 47711. Games are
MULTIDIE(dice roller),DUNGEON GRAFIX I,DUNGEON GRAFIX II, and FANTASY PEOPLE
for advanced players.$6 on your C-30,or $8 on his. no listings.

BOWLING,No. of pins hit and score shown $5 on your tape or $6 on his. Add $2
and either HANGMAN or CHECKERS will be put on other side. Listing for half price.
John Collins,713 Bradford Dr. Ft. Walton Beach,FL 32548.

Bob Weber is part of W&W Software Sales,6594 Swartout Rd. Algonac, MI 48001
and has reorganized their software. They now have 4 cassettes with 5 programs
each for $10 each in a preprogrammed form. Send for complete list/descriptions.

<u>PROGRAMMABLE KEYBOARD?</u> The latest delay hinges on a Texas Instruments petition now
before the FCC. They have asked for a change in the procedures by which computer/parts
are approved for use. Apparently RCA tried this two years ago but never pushed very
hard. The current procedure is for a computer manufacturer to have his whole system
approved as meeting RFI, TVI, etc., limits, while the proposal is to have only the
RF modulator inspected/approved. This has thrown the manufacturers into a tizzy as
they feel that if the rules or standards are changed in the middle of the stream
they'll have design and manufacturing costs to contend with. Bally, who haven't gone
into production yet, feels they have to wait and see which way the FCC goes to avoid
the 50-50 chance of being in error when the decision is made. For a company that
thrives on gambling, those are unacceptable odds. Here is an excerpt from <u>Consumer
Electronics</u> monthly 5/79 p. 31:

Doubt about future standards for personal computers will delay product introductions as well as halt production of units introduced, but not yet in production. Bally, for one, will hold off manufacturing its upgraded game/computer introduced at Winter CES until a decision is reached.

"We have two units to go with," says national sales manager, Jack Nieman, "the one we showed at CES for around $650 and a lower-priced unit for around $350. But it could cost us millions of dollars if we make a decision on which unit to go with before the FCC makes a decision." □

My discussions with Bally indicate that they expect a corporate discussion/decision
around January, with delivery months later than that. Needless to say, the whole
situation is disappointing and frustrating. Fortunately, we have some people work-
ing on ways to 'make our own', and we can see a little glimmer of light, as reported
earlier this issue. Subscribers who are working on a keyboard,memory addition, or
any other "goodie" are urged to write me so that I can build up a team to get the
needed hardware built and available for us all.

<u>ARCADIAN</u>

Robert Fabris, stapler
3626 Morrie Dr.
San José,CA 95127

SAN JOSE, CA
JUN 18
PM
1979

USA 15c

94550HUS0635L
R. HAUSER
635 LOS ALAMOS AVENUE
LIVERMORE, CA   94550

FIRST CLASS